MOTOROLA

MEX68PP3(D)

MEX68PP3
PROM PROGRAMMER III
User's Guide

MICROSYSTEMS

MEX68PP3(D)
APRIL 1978

Se mettre en simple MAP

MEX68PP3

PROM PROGRAMMER III MODULE SUPPLEMENT
M6800 EXORciser User's Guide

The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the product described any license under the patent rights of Motorola Inc. or others.

Motorola reserves the right to change specifications without notice.

EXORciser, EXbug, MINIbug II/III, and MICRObug are trademarks of Motorola Inc.

Second Edition
Copyright 1978 by Motorola Inc.
First Edition November 1977

TABLE OF CONTENTS

CHAPTER		Page
1.	GENERAL INFORMATION	1-1
1.1	Introduction	1-1
1.2	Features	1-1
1.3	Specifications	1-1
1.4	General Description	1-1
1.5	Equipment Supplied	1-3
2.	INSTALLATION INSTRUCTIONS, PREPARATION FOR USE, AND INTERCONNECTION CONSIDERATIONS	2-1
2.1	Introduction	2-1
2.2	Unpacking Instructions	2-1
2.3	Inspection	2-1
2.4	Module Interconnections	2-1
2.5	Installation and Removal	2-1
3.	OPERATING PROCEDURES	3-1
3.1	Introduction	3-1
3.2	Load and Start-Up Procedures	3-1
3.3	Programmer Commands and Command Format	3-2
3.4	Specifying EROM/PROMs	3-4
3.5	Load and Display Memory	3-5
3.6	Program, Display and Verify EROM/PROM	3-6
4.	THEORY OF OPERATION	4-1
4.1	Introduction	4-1
4.2	General Description	4-1
4.3	Block Diagram Description	4-1
5.	PROGRAMMER MODULE COMPONENTS	5-1
APPENDIX A:	PROGRAMMING EXAMPLES	A-1
A-1:	Program EROM/PROM from Tape or Disk	A-2
A-2:	Duplicating a PROM	A-3
A-3:	Changing the Unprogrammed Bits of a PROM	A-4
A-4:	Possible Error Modes	A-5,6
APPENDIX B:	Error Messages	B-1,2
APPENDIX C:	Adapting Programmer Software to Your System	C-1/C-5
APPENDIX D:	Program Flow Description	D-1
APPENDIX E:	Memory Maps	E-1

LIST OF ILLUSTRATIONS

Figure		Page
1-1	MEX68PP3 PROM Programmer III Module	1-2
1-2	PROM Programmer III Module Memory Map	1-5
4-1	PROM Programmer III Block Diagram	4-4
4-2	PROM Programmer III Schematic	4-5,6
5-1	PROM Programmer III Module Component Locations	5-4

LIST OF TABLES

Tables		Page
1-1	MEX68PP3 PROM Programmer III Module Specifications	1-3
1-2	EROM/PROM Devices Recognized by the PROM Programmer III Module	1-4
2-1	Bus Interface Signals	2-2
3-1	Command Summary	3-3
3-2	PROM Programmer III Control Commands	3-4
5-1	PROM Programmer III Module Components	5-1,2,3

CHAPTER 1

GENERAL INFORMATION

1.1 INTRODUCTION

This manual is a supplement to the M6800 EXORciser User's Guide. It provides general information, preparation for use and installation instructions, operating procedures, and theory of operation for the MEX68PP3 PROM Programmer III Module. Appendices provided at the rear of this supplement provide typical programming examples, error messages, and software modifications for adapting the Programmer to a user system. A typical PROM Programmer III Module is shown in Figure 1-1. All address references within this manual are shown in hexadecimal unless otherwise indicated.

1.2 FEATURES

The features of the MEX68PP3 PROM Programmer III Module include:

- 57 different EROMs and PROMs can be programmed
- Compatible with EXbug, MINIbug II, MINIbug III, and MICRObug firmware programs
- Adaptable to user-prepared programs
- Transfers data from programmed device to system memory for duplication or verification
- Programs EROMs and PROMs from system memory
- Verifies data in programmed device
- Jumper-selectable address relocation
- Operating software available on three different mediums: cassette, paper tape, or diskette (either EDOS or MDOS versions)
- EXORciser and Micromodule Family bus compatible

1.3 SPECIFICATIONS

The MEX68PP3 PROM Programmer III Module specifications are identified in Table 1-1.

1.4 GENERAL DESCRIPTION

The MEX68PP3 PROM Programmer III Module provides the necessary hardware and software to program 57 different EROM and PROM devices, verify the data in the programmed device, and transfer data from the device to the system RAM memory. The Programmer will plug directly into any EXORciser or Micromodule-based system chassis.

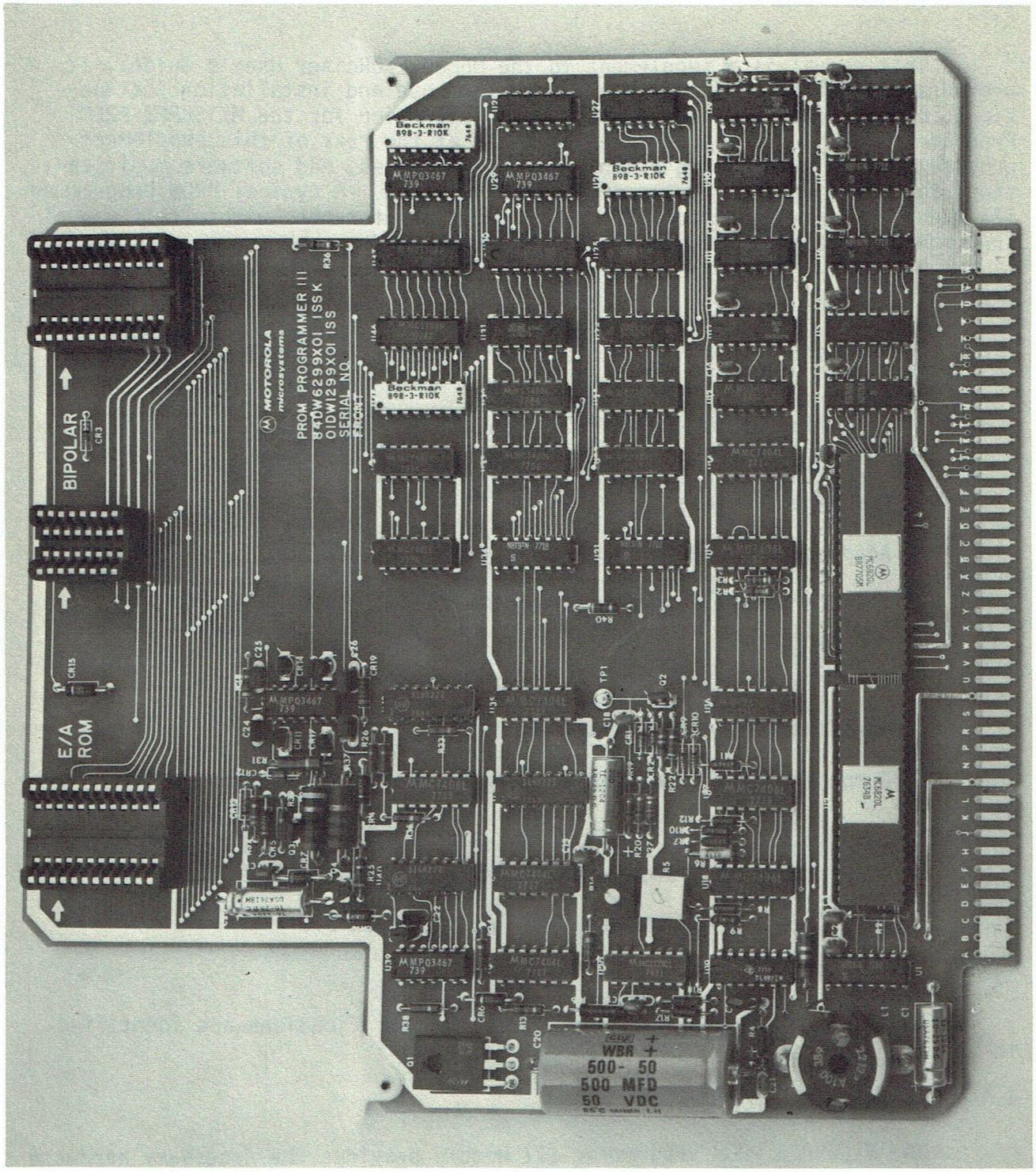


FIGURE 1-1. MEX68PP3 PROM Programmer III Module

TABLE 1-1. MEX68PP3 PROM Programmer III Module Specifications

CHARACTERISTICS	SPECIFICATIONS
Power Requirements	+5 Vdc @ 700 mA +12 Vdc @ 200 mA -12 Vdc @ 50 mA
Input Signals	TTL voltage compatible
Output Signals	Three-state TTL voltage compatible
Physical Characteristics	
Width	9.75 in.
Height	7.50 in.
Board Thickness	0.062 in.
Programs EROM/PROM devices	See Table 1-2 for listing of EROM/PROM devices

Software for the PROM Programmer III Module is available in any of three different mediums: cassette, paper tape, or diskette (EDOS or MDOS versions). The diskette versions are compatible only with the EXORciser, since both require the EXbug firmware program to operate the respective disk system. The cassette and paper tape versions, however, are compatible with EXbug, MINIbug II, MINIbug III, and MICRObug firmware routines. The software provided with the Programmer can also be modified for use with programs prepared by the user; the only restriction is that input/output subroutines in the user-prepared program must be available for accessing by the PROM Programmer III Module software (refer to Appendix C).

As shipped, the Programmer resides at addresses EC08 through EC0F. A jumper connection on the module permits address relocation to addresses BC08 through BC0F. A memory map is provided in Figure 1-2 showing the system memory requirements for all software versions. Appendix D provides a description of program flow.

1.5 EQUIPMENT SUPPLIED

The PROM Programmer II Module is shipped with its software on digital cassette (option MEX68PP3A), paper tape (option MEX68PP3B), or diskette (option MEX68PP3D for EDOS or MEX68PP3M for MDOS). Also included is this supplement to the M6800 EXORciser User's Guide.

TABLE 1-2. EROM/PROM Devices Recognized by the PROM Programmer III Module

MANUFACTURER	PART NO.	DESCRIPTION
MOTOROLA		
	MCM7640	512X8 BIPOLAR
	MCM7641	512X8 BIPOLAR
	MCM7680	1KX8 BIPOLAR
	MCM7681	1KX8 BIPOLAR
	MCM68708	1KX8 EROM
	MCM2716	2KX8 EROM
INTEL		
	INT2704	512X8 EROM
	INT2708	1KX8 EROM
7FF	-INT2716	2KX8 EROM
	INT2758	1KX8 EROM
TEXAS INSTRUMENTS		
	TMS2708	1KX8 EROM
	TMS2716	2KX8 EROM
MONOLITHIC MEMORIES		
	MMI5300	256X4 BIPOLAR
	MMI5301	256X4 BIPOLAR
	MMI6300	256X4 BIPOLAR
	MMI6301	256X4 BIPOLAR
	MMI5305	512X4 BIPOLAR
	MMI5306	512X4 BIPOLAR
	MMI6305	512X4 BIPOLAR
	MMI6306	512X4 BIPOLAR
	MMI5335	256X8 BIPOLAR
	MMI5336	256X8 BIPOLAR
	MMI6335	256X8 BIPOLAR
	MMI6336	256X8 BIPOLAR
	MMI5340	512X8 BIPOLAR
	MMI5341	512X8 BIPOLAR
	MMI6340	512X8 BIPOLAR
	MMI6341	512X8 BIPOLAR

MANUFACTURER	PART NO.	DESCRIPTION
MONOLITHIC MEMORIES		
	MMI5380	1KX8 BIPOLAR
	MMI5381	1KX8 BIPOLAR
	MMI6380	1KX8 BIPOLAR
	MMI6381	1KX8 BIPOLAR
	MMI5384	1KX8 BIPOLAR
	MMI5385	1KX8 BIPOLAR
	MMI6384	1KX8 BIPOLAR
	MMI6385	1KX8 BIPOLAR
HARRIS		
	HM-7610	256X4 BIPOLAR
	HM-7611	256X4 BIPOLAR
	HM-7620	512X4 BIPOLAR
1FF	-HM-7621	512X4 BIPOLAR
	HM-7640	512X8 BIPOLAR
	HM-7641	512X8 BIPOLAR
3FF	-HM-7644	1KX4 BIPOLAR
	HM-7680	1KX8 BIPOLAR
	HM-7681	1KX8 BIPOLAR
SIGNETICS		
	82S126	256X4 BIPOLAR
	82S129	256X4 BIPOLAR
	82S130	512X4 BIPOLAR
	82S131	512X4 BIPOLAR
	82S140	512X8 BIPOLAR
	82S141	512X8 BIPOLAR
	82S180	1KX8 BIPOLAR
	82S181	1KX8 BIPOLAR
	82S190	2KX8 BIPOLAR
	82S191	2KX8 BIPOLAR
	82S2708	1KX8 BIPOLAR
	SIG2708	1KX8 EROM

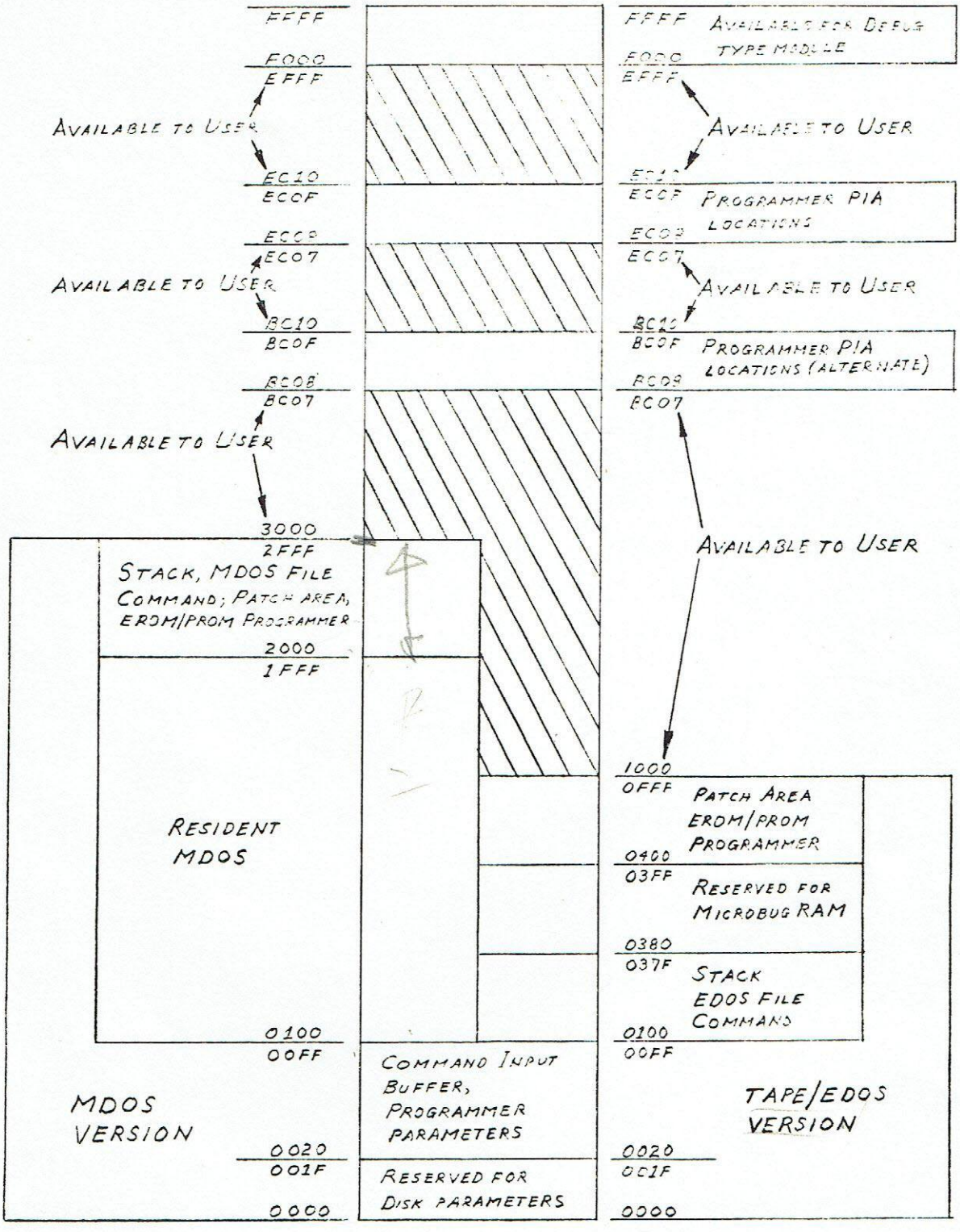


FIGURE 1-2. PROM Programmer III Module Memory Map

CHAPTER 2

INSTALLATION INSTRUCTIONS, PREPARATION FOR USE, AND INTERCONNECTION CONSIDERATIONS

2.1 INTRODUCTION

This chapter provides the unpacking, inspection, installation, and removal procedures for the MEX68PP3 PROM Programmer III Module. This chapter also discusses the module's interconnection signals.

2.2 UNPACKING INSTRUCTIONS

The MEX68PP3 PROM Programmer III Module may be received as part of an M6800 EXORciser or as an individual unit. If the Programmer Module has been shipped as part of an EXORciser, refer to the M6800 EXORciser User's Guide for unpacking instructions. When the Programmer is shipped as an individual unit, unpack in accordance with the following paragraph.

Unpack the MEX68PP3 PROM Programmer III Module from its shipping carton and, referring to the packing list, verify that all of the items are present. Save the packing materials for storing and reshipping of the module. If the shipping carton is damaged upon receipt, request that the carrier's agent be present while the module is being unpacked and inspected.

2.3 INSPECTION

The MEX68PP3 PROM Programmer III Module should be inspected upon receipt and at periodic intervals for loose, broken or missing components, or physical damage to the printed circuit board. When not in use, the Programmer Module should be stored in a clean, moisture-free location.

2.4 MODULE INTERCONNECTIONS

The Programmer Module interconnects directly with either the EXORciser or Micromodule Family bus. The bus signals are listed in Table 2-1. This table lists each pin connection, signal mnemonic, and signal characteristic.

2.5 INSTALLATION AND REMOVAL

Install the Programmer Module in the following manner:

A. Remove system power.

CAUTION

DAMAGE TO THE COMPONENTS ON THE MODULE MAY RESULT IF INSTALLED WHILE POWER IS APPLIED.

B. Install the Programmer Module into the selected card slot. If the module is installed in an EXORciser, one of the six outside card slots (three on each side of the card cage) should be used.

C. Apply system power.

D. Program the device(s) in accordance with the procedures provided in Chapter 3.

E. After all programming has been completed, turn off the system power and remove the Programmer Module. This prevents the possibility of the Programmer Module inadvertantly responding to instructions being executed during program development.

TABLE 2-1. Bus Interface Signals

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
A,B,C	+5 V	+5 Vdc - Used for module's logic circuits.
D to H		Not Used
J	Ø2	PHASE 2 - Input clock signal.
K,L		Not Used
M	-12 V	-12 Vdc - Used to power the programming-related circuits.
N to S		Not Used
T	+12 V	+12 Vdc - Used to power the programming-related circuits.
U to Z		Not Used
$\overline{A}$ to $\overline{F}$		Not Used
$\overline{H}$	$\overline{D3}$	DATA bus (bit 3) - One of 8 bi-directional lines that, when enabled, provides a two-way data transfer between this module and the rest of the system.
$\overline{J}$	$\overline{D7}$	DATA bus (bit 7) - Same as $\overline{D3}$ on P1-$\overline{H}$.
$\overline{K}$	$\overline{D2}$	DATA bus (bit 2) - Same as $\overline{D3}$ on P1-$\overline{H}$.
$\overline{L}$	$\overline{D6}$	DATA bus (bit 6) - Same as $\overline{D3}$ on P1-$\overline{H}$.
$\overline{M}$	A14	ADDRESS bus (bit 14) - One of 16 input address lines used to select a specific Programmer operation.
$\overline{N}$	A13	ADDRESS bus (bit 13) - Same as A14 on P1-$\overline{M}$.
$\overline{P}$	A10	ADDRESS bus (bit 10) - Same as A14 on P1-$\overline{M}$.

TABLE 2-1. Bus Interface Signals (Contd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
$\overline{R}$	A9	ADDRESS bus (bit 9) - Same as A14 on P1-$\overline{M}$.
$\overline{S}$	A6	ADDRESS bus (bit 6) - Same as A14 on P1-$\overline{M}$.
$\overline{T}$	A5	ADDRESS bus (bit 5) - Same as A14 on P1-$\overline{M}$.
$\overline{U}$	A2	ADDRESS bus (bit 2) - Same as A14 on P1-$\overline{M}$.
$\overline{V}$	A1	ADDRESS bus (bit 1) - Same as A14 on P1-$\overline{M}$.
$\overline{W}$,$\overline{X}$,$\overline{Y}$	GND	GROUND
1,2,3	+5 V	+5 Vdc - Used for module's logic circuits.
4		Not Used.
5	$\overline{\text{RESET}}$	RESET - This buffered input signal permits this module to be restarted by either an external switch closure to ground or by the low level RESET signal from the module containing the MPU.
6	R/W	READ/WRITE - This signal controls whether data is to be read (high) from the module or written (low) into the module. The normal standby state of the signal is read (high). This signal will also be in the read state whenever the MPU is halted.
7 to 9		Not Used.
10	VUA	VALID USER'S ADDRESS - This high level signal is used to indicate to the module that the address on the address bus is valid for system use and is not required by the debug device (EXbug, MINIbug II/III, or MICRObug).
11	-12 V	-12 Vdc - Used to power the programming-related circuits.
12 to 15		Not Used.
16	+12 V	+12 Vdc - Used to power the programming-related circuits.
17 to 28		Not Used.
29	$\overline{D1}$	DATA bus (bit 1) - Same as $\overline{D3}$ on P1-$\overline{H}$.

TABLE 2-1. Bus Interface Signals (Contd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
30	$\overline{D5}$	DATA bus (bit 5) - Same as $\overline{D3}$ on P1-$\overline{H}$.
31	$\overline{D0}$	DATA bus (bit 0) - Same as $\overline{D3}$ on P1-$\overline{H}$.
32	$\overline{D4}$	DATA bus (bit 4) - Same as $\overline{D3}$ on P1-$\overline{H}$.
33	A15	ADDRESS bus (bit 15) - Same as A14 on P1-$\overline{M}$.
34	A12	ADDRESS bus (bit 12) - Same as A14 on P1-$\overline{M}$.
35	A11	ADDRESS bus (bit 11) - Same as A14 on P1-$\overline{M}$.
36	A8	ADDRESS bus (bit 8) - Same as A14 on P1-$\overline{M}$.
37	A7	ADDRESS bus (bit 7) - Same as A14 on P1-$\overline{M}$.
38	A4	ADDRESS bus (bit 4) - Same as A14 on P1-$\overline{M}$.
39	A3	ADDRESS bus (bit 3) - Same as A14 on P1-$\overline{M}$.
40	A0	ADDRESS bus (bit 0) - Same as A14 on P1-$\overline{M}$.
41,42,43	GND	GROUND

CHAPTER 3

OPERATING PROCEDURES

3.1 INTRODUCTION

The PROM Programmer III Module software, along with the companion hardware module, provides the means to program a variety of 4-bit and 8-bit EROMs and PROMs. The devices can be programmed and their data verified, stored in system memory, and retrieved for duplication. This chapter describes software load and start procedure, provides programmer commands and their formats, and supplies the procedures to select a specific EROM/PROM device load and display and load, display and verify an EROM/PROM.

Typical programming examples, error message descriptions, and modifications to the software can be found in appendices at the end of this supplement.

3.2 LOAD AND START-UP PROCEDURES

The EDOS version of the software is loaded into memory from the EDOS file PROMP, using the appropriate EDOS command to load a file. (Refer to EDOS User's Guide.) The tape version should be loaded into memory following the directions for loading tape with the monitor in use.

The cold start execution address for both the tape and EDOS versions is hexadecimal 404. After the program has been loaded into memory, it is started by transferring control to this address, using the appropriate monitor command. Both the tape and EDOS versions have a warm start address of hexadecimal 407. This entry point can be used to re-enter the program and retain the current parameters. This is useful after the EXIT command has been used to return to the monitor to change a memory location.

The MDOS version is treated as an MDOS supported command. It is loaded and started by typing PROMPROG as an MDOS command. After the EXIT command has been executed in the MDOS version, the MAID command ;P will warm start the programmer. MDOS should NOT be restarted by the MAID command E800;G, following the EXIT command, until after an ABORT or RESTART. MDOS can be reentered by using the programmer's DISK command.

When the programmer is entered at the cold start entry point, it responds by printing:

```
PROM PROGRAMMER j.k
COPYRIGHT MOTOROLA vvvv
P
```

where j.k is the programmer version and revision number, vvvv is the copyright year, and P is the programmer's prompt. THE PROGRAMMER MUST BE STARTED INITIALLY AT THE COLD START ENTRY POINT.

When the program is re-entered at the warm start entry point, it responds by displaying only the prompt for the mode it is in. Table 1-2 lists the part numbers recognized by the Programmer. The EROM/PROM part number must be entered exactly as listed. The first three characters of the part number are used to identify the part's manufacturer. The remainder of the part number identifies the part size and type.

EROM/PROMs may be inserted or removed any time the Programmer is displaying its prompt. EROM/PROMs should not be removed or inserted while a command is being executed. IN ALL CASES THE EROM/PROM MUST BE INSERTED IN THE PROPER SOCKET AS MARKED ON THE BOARD.

3.3 PROGRAMMER COMMANDS AND COMMAND FORMAT

All commands are of the format:

<cmnd><delim><params><term>

where:

<cmnd>	is an EROM/PROM part number or a four letter command
<delim>	is a command line delimiter, either a comma or a space
<params>	are the parameters required for the command, if any. Multiple parameters are separated by commas. If a null parameter value is entered for any required parameter, the last value of that parameter will be used.
<term>	is the command line terminator, a carriage return. No action is taken on the command until the carriage return is entered.

Until the carriage return is entered, a rubout will delete the last command line character and cause it to be typed. A control-X character will delete the entire line and cause a prompt to be issued. A control-D character will display what has been entered and not deleted.

The commands that print from memory or EROM/PROM will suspend printing when a control-W is entered. Once stopped, any character other than control-W will cause the command to continue. The BREAK key will cause these commands to abort.

The PROM Programmer Module prompt is the letter P followed by a space. If a 4-bit EROM/PROM device is to be programmed and the upper or lower four bits of memory have been specified, the prompt will be preceded by "<" or ">", respectively.

In the command descriptions, hexadecimal numeric parameters are indicated by a string of three or four alphabetic characters -- for example, xxxx. If a parameter is not entered, the current value of that parameter is used. The value of a parameter is saved each time the parameter is entered as part of a valid command. All parameters are initialized to zero on Programmer start-up. All commands are terminated by a carriage return.

Table 3-1 lists the Programmer commands and their descriptions. Command parameters do not have to be entered if the parameter is already the desired value. The EROM/PROM part numbers listed in Table 1-2 are accepted as commands, in order to specify the EROM/PROM to be used.

TABLE 3-1. Command Summary

COMMAND	DESCRIPTION
CHCK aaa,zzz	Check aaa words of the EROM/PROM starting at EROM/PROM offset zzz for the unprogrammed state.
DISK	Reenter the disk operating system (EDOS and MDOS versions only).
EXIT	Exit to the system monitor.
FILE filename,nnnn	Load the disk file filename into memory, adding the offset nnnn to the load address.
LS4B	Use the least significant four bits of memory with four-bit PROMs. Default condition when a four-bit PROM is programmed.
MS4B	Use the most significant four bits of memory with four-bit PROMs.
OFST nnnn	Set the memory offset parameter nnnn to the value entered.
PROG xxxx,yyyy,zzz	Program the EROM/PROM starting at EROM/PROM offset zzz from memory locations xxxx+nnnn through yyyy+nnnn.
READ xxxx,yyyy,zzz	Read the EROM/PROM to memory addresses xxxx+nnnn through yyyy+nnnn starting at EROM/PROM offset zzz.
SHWM xxxx,yyyy	Show memory locations xxxx+nnnn through yyyy+nnnn in both hex and ASCII forms.
SHWP aaa,zzz	Show the EROM/PROM contents in hex form. The ASCII form is also displayed if an eight-bit PROM is being shown.
STAT	Display the Programmer's status.
TAPE nnnn	Load from the system console tape reader to memory adding the offset nnnn to the load address.
VERF xxxx,yyyy,zzz	Verify the EROM/PROM starting at PROM offset zzz against memory locations xxxx+nnnn through yyyy+nnnn.

Table 3-2 lists and further describes those commands that permit the user to display the Programmer status, change the memory offset parameter nnnn, and exit the Programmer.

TAB:E 3-2. PROM Programmer III Control Commands

COMMAND	DESCRIPTION
DISK	This command exits the Programmer program and restarts the disk operating system. The DISK command is valid only for the disk versions of the PROM Programmer III Module.
EXIT	This command exits the Programmer program and returns to the system monitor. The tape/EDOS version can be warm started (parameters retained) by restarting the Programmer program at hex address 407. The MDOS version can be warm started by entering the MAID command ;P.
OFST nnnn	This command changes the memory offset nnnn to the value entered. The memory offset parameter nnnn is used in the FILE, READ, TAPE, PROG, SHWP, and VERF commands. The TAPE and FILE commands can also change it.
STAT	This command displays the current Programmer status including the current EROM/PROM specification, if any. The parameters xxxx, yyyy, nnnn, aaa, and zzz are displayed. If a four-bit PROM is specified, the indication of the bits of memory to be used in READ, PROG, and VERF commands is included in the prompt.

3.4 SPECIFYING EROM/PROMs

To specify the EROM/PROM to be used, the EROM/PROM part number (MCM68708, for example) is entered at the console, followed by a carriage return. A list of valid EROM/PROM devices is contained in Table 1-2.

Specifying an EROM/PROM updates the parameter aaa to the number of words in the EROM/PROM. If any error message is printed when attempting to specify an EROM/PROM, the new EROM/PROM specification has not been accepted. The commands that interface to the EROM/PROM (CHCK, READ, SHWP, PROG, and VERF) cannot be executed until an EROM/PROM has been specified. Additionally, if a 4-bit PROM is specified, the following two commands specify the four bits of memory to be used by the READ, PROG, and VERF commands. These two commands are valid only when a 4-bit PROM is specified. When a 4-bit PROM is first specified, the default mode is LS4B.

LS4B — This command specifies that the least significant four bits are to be used during the PROG, READ, and VERF commands. While in this mode, the prompt is preceded by ">".

MS4B — This command specifies that the most significant four bits of memory are to be used during READ, PROG, and VERF commands. While in this mode, the prompt is preceded by "<".

3.5 LOAD AND DISPLAY MEMORY

The following commands provide ways to load and display memory.

FILE filename,nnnn — This command loads the disk file specified, adding the offset nnnn to the load address. This command is valid only on the disk versions of the program; filename must be in the correct file name and drive number format for the disk system in use. The default drive number is zero. The MDOS version default suffix is LO. This command sets the parameter xxxx equal to the lowest load address of the file. If an EROM/PROM is specified, the parameter yyyy is set equal to the lowest load address of the file plus the EROM/PROM size minus one. If no EROM/PROM is specified, yyyy is unchanged. The command stops loading when the end of file is reached. The MDOS version FILE command does not verify that memory is changing as it is being loaded. The EDOS version does. The EDOS file must be an S record object file. The MDOS file must be a memory image file. The EDOS version FILE command loads to the end of the file, ignoring S9 records.

READ xxxx,yyyy,zzz — This command reads from EROM/PROM starting offset zzz to memory addresses xxxx+nnnn through yyyy+nnnn. The current value of the memory offset nnnn is used. When reading a four-bit PROM, only the memory bits specified (least or most significant four bits) will be changed.

SHWM xxxx,yyyy — This command shows the memory from xxxx+nnnn through yyyy+nnnn in both hex and ASCII forms.

TAPE nnnn — This command loads an S record object tape to memory from the console tape reader, adding the offset nnnn to the load address. The parameter xxxx is updated to the lowest load address of the tape. If an EROM/PROM has been specified, yyyy is updated to the lowest load address plus the EROM/PROM size minus one; otherwise, yyyy is unchanged. Due to the way the TI cassette reader is controlled at 1200 and 2400 baud, the record following the S9 record on tape will be skipped. The record following a load error at 1200 and 2400 baud will also be skipped. The command stops loading when the S9 record is read or an end of tape condition is detected by not receiving characters (timeout).

3.6 PROGRAM, DISPLAY AND VERIFY EROM/PROM

The following commands are used to program, show, and verify an EROM/PROM. These commands will not be executed unless an EROM/PROM has been specified.

CHCK aaa,zzz

This command checks aaa words in the EROM/PROM beginning at offset zzz for the unprogrammed state. The EROM/PROM offset and contents of all programmed locations are printed.

PROG xxxx,yyyy,zzz

This command programs the EROM/PROM from memory locations xxxx+nnnn through yyyy+nnnn beginning at EROM/PROM offset zzz. Before attempting to program the EROM/PROM, the Programmer verifies that the +5 volt supply to bipolar PROMs is at a logic '1' level. If it is not, the message "PWR?" is printed and the command is aborted. This condition can occur if the PROM is defective or is inserted upside down. Next the Programmer checks that the portion of the EROM/PROM to be programmed is unprogrammed. If it is not programmed, the Programmer will attempt to program the EROM/PROM. If it is programmed, the Programmer will print the message "CHNG?" and wait for the user to respond. Typing the letter Y will cause the Programmer to attempt to program the EROM/PROM. Typing any other character will cause the command to abort. If the Programmer is unable to program a bit in a bipolar PROM, it will print the message "NO CHNG mmmm pp" and then abort the command; mmmm is the PROM location that would not program and pp is the contents of that location. On completion of programming, the portion of the EROM/PROM programmed is verified against memory. The EROM/PROM offset and EROM/PROM and memory contents of all locations that do not compare are printed. The PROG command can be aborted by the terminal "BREAK" key while verify errors are being printed. At all other times, the PROG command should not be interrupted. However, if it is necessary to stop the command while a part is being programmed, a system reset or restart should be used so that the PIA controlling the programming voltage will be reset, thus removing the programming voltage from the part.

SHWP aaa,zzz

This command prints in hex on the console, at least aaa words beginning at the EROM/PROM offset zzz. If an eight-bit EROM/PROM is specified, the ASCII format will also be displayed. The display will start at the first 16-word boundary less than or equal zzz (effectively resetting the four least significant bits of the start EROM/PROM offset for the display). The number of words shown will be aaa, rounded up to the next 16-word boundary greater than or equal aaa.

VERF xxxx,yyyy,zzz This command verifies the EROM/PROM with memory locations xxxx+nnnn through yyyy+nnnn, starting at EROM/PROM offset zzz. The EROM/PROM offset, and EROM/PROM and memory contents of each location that does not verify, are printed.

CHAPTER 4

THEORY OF OPERATION

4.1 INTRODUCTION

This chapter provides a hardware description of the MEX68PP3 PROM Programmer III Module. Figures 4-1 and 4-2 are the block diagram and schematic of the module.

4.2 GENERAL DESCRIPTION

The Programmer Module provides the means to program a variety of 4-bit and 8-bit memory devices. The Module is compatible to the EXORciser and Micromodule bus; the Programmer software is compatible to EXbug I, MINIbug II, MINIbug III, and MICRObug monitors, and can be adapted to others. The module allows for device programming, verification, data transfer to system memory, and retrieval for duplication.

Software for the Programmer is available in two versions. One is a tape/EDOS version offered on digital cassette, paper tape, or diskette. This version is compatible with EXbug I, MINIbug II, MINIbug III, and MICRObug. It may also be adapted for use with other monitors that make input/output subroutines available to the user.

An MDOS version, available on diskette only, is intended for use on an EXbug/MDOS system. This version does not accept command input from an MDOS CHAIN file; it will only accept command input from the console. The MDOS version can also be used with other monitors.

Both the tape/EDOS version and the MDOS version allow MPU clock rates of up to 2 MHz and permit the Programmer to be located at addresses EC08 through ECOF or BC08 through BCOF. Both versions must be modified to work with monitors other than EXbug 1 and when clock rates are $\leq$0.7 MHz or >1.0 MHz. (See Appendix C for the required modification.)

4.3 BLOCK DIAGRAM DESCRIPTION

As shown in Figure 4-1, a 24-pin DIP socket is used to program 8-bit wide EROM devices. A 16-pin socket and a 24-pin socket are used to program 4-bit wide and 8-bit wide bipolar PROMs, respectively. The EROM devices are programmed one word (8 bits) at a time; the bipolar devices are programmed one bit at a time. Each category of device requires unique timing and syntax (order relationship) considerations; these requirements are met by means of the Programmer software alone.

The current and voltage required to program an EROM or PROM differ from the normal operating levels. The current required to program a Harris (HAR) or Motorola (MOTO) PROM (or equivalent) is supplied via the Vcc pin of the device. The current to program a Monolithic Memories, Inc. (MMI) PROM (or

equivalent) is supplied via the programming pin of the device. When a Signetics (SIG) PROM (or equivalent) is being programmed, the required current is supplied via the data pins. The Vcc, programming, and/or the data pins of these various devices are driven to higher voltage levels during programming by circuits on the Programmer Module.

A linear voltage regulator supplies Vcc to the bipolar PROMs. The regulator also supplies (via the Vp bus of the bit readback circuit) resistors used as loads during bit programming and as loads for open collector-type PROMs. Under Programmer software control, the output of the regulator is switched between one of three voltage levels. At all times other than bit programming, Vcc and Vp are nominally +5.0 V. When an MMI device is being programmed, control signal MMI' goes to logic level "1" and Vcc and Vp increase to +5.5 V, nominal, during the programming attempt. When a SIG device is programmed, control signal SIG' goes to logic level "1" and Vcc and Vp increase to +8.75 V, nominal, during the programming attempt.

A DC to DC converter supplies the program pin drive circuit and the data bit driver Vpp bus (routed through a generic code interface). The converter output is also connected to the PROM sockets Vcc pins, to provide a current boost, when HAR or MOTO devices are programmed. When a HAR device is being programmed, control signal $\overline{\text{HAR+MOTO}}$' goes to logic level "1" during bit programming, causing Vcc and Vp to increase to +12 V, nominal. When a MOTO device is being programmed, control signals $\overline{\text{HAR+MOTO}}$' and MOTO go to logic level "1", causing Vcc and Vp to increase to +15 V, nominal, during bit programming.

Three other output voltage levels are available from the DC to DC converter, under Programmer software control. When a SIG device is being programmed, control signals SIG and SIG' go to logic level "1"; the output of the converter is gated through the generic code interface and applied to the Vpp bus of the data bit drivers. The Vpp bus ramps up to a nominal +17 V. When enabled by the $\overline{\text{STROB}}$' control signal going to logic level "1", the data bit drivers complete the path of current to program a SIG device. When an MMI device is being programmed, control signal MMIP goes to logic level "0" while control signal MMI' goes to logic level "1". The output of the DC to DC converter ramps up to +27 V, nominal. The converter output is gated through the program pin drive circuit to the program pins of the PROM sockets to complete the path of current to program an MMI device. At this time, the converter output is also gated through the generic code interface to supply the bit drivers Vpp bus with a nominal +20 V.

The fifth output voltage level of the DC to DC converter is used during EROM programming. When control signal EROMPGEN' goes to logic level "1", the output of the converter is gated through the program pin drive circuit to the program pin of the EROM socket. The voltage level at the EROM program pin is +26 V, nominal.

The data bit drivers are capable of supplying up to 200 mA at 20 V, nominal, to current-mode PROMs. The data bit driver circuit also provides the on and off potentials to program voltage mode PROMs. Selection of a 0V/5V hold off potential for unprogrammed data bits is made by a V hold off

select circuit, controlled by the CB2 output of (Peripheral Interface Adapter) PIA#1. When control signal $\overline{RD\ BK}$ goes to logic level "0", the PROM data lines ID0 through ID7 are read back to the EROM/PROM data lines PD0 through PD7 for programming verification.

This Programmer configures the E/A ROM socket according to the following table.

	PROGRAM					READ				
Pin No.	18	19	20	21	24	18	19	20	21	24
INT2758	TTL_P	TTL_L	$\overline{CS}_H$	+25	+5	TTL_L	TTL_L	$\overline{CS}_L$	+5	+5
INT2716	TTL	A10	$\overline{CS}_H$	+25	+5	TTL_L	A10	$\overline{CS}_L$	+5	+5
INT2708	+26	+12	+12	-5	+5	0	+12	+5	-5	+5
TMS2716	+26	+12	A10	-5	+12	0	+12	A10	-5	+5

TTL_P = TTL Pulsed Low to High
$\overline{CS}_H$ = TTL High
$\underline{CS}_L$ = TTL Low
A10 = TTL as required

In an M6800-based system, each PIA requires four locations in memory. The PIAs of the Programmer Module are located at hexadecimal addresses EC08 through EC0F (PIA#1:EC08-EC0B;PIA#2:EC0C-EC0F). By grounding a signal line in the Module's address bus interface, the PIAs can be relocated to hexadecimal address BC08 through BC0F. Two wire wrap pins identified as J1 are provided for this purpose.

The address bus interface circuit buffers the system address lines A0 through A15. Bits A0' to A2' are used for PIA register and partial chip selection. Bits A3', A4'-A9' and A10'-A15', along with control signal VUA (Valid User's Address) are used by the I/O Address Decoder and Control circuit to complete the chip selection. The I/O Address Decoder and Control circuit also uses the Ø2, R/W' and $\overline{R/W}$' signals, from the Control Bus Interface buffers, to generate the READ and $\overline{WRITE}$ control signals for the Data Bus Interface. The Data Bus Interface consists of three-state transceivers for the $\overline{D0}$-$\overline{D7}$ data bus. The transceivers are in the high impedance state unless the Programmer Module has been selected by the system, via the address and control buses. During an MPU read operation, the R/W control signal goes to a logic "1" level. If the Programmer Module has been selected, the Data Bus Interface inverting drivers will be enabled and the PIA D0'-D7' bits will be read from the $\overline{D0}$-$\overline{D7}$ bus. During an MPU write operation, the R/W signal goes to logic level "0". If the Programmer Module has been selected, the Data Bus Interface receivers will be enabled and data from the $\overline{D0}$-$\overline{D7}$ bus will be written into the selected PIA.

The Control Bus Interface also buffers the system $\overline{RESET}$ control signal. When an active low $\overline{RESET}$ pulse occurs (during system power-on or restart operations), the registers in the PIAs are reset low; the PIAs are then reconfigured.

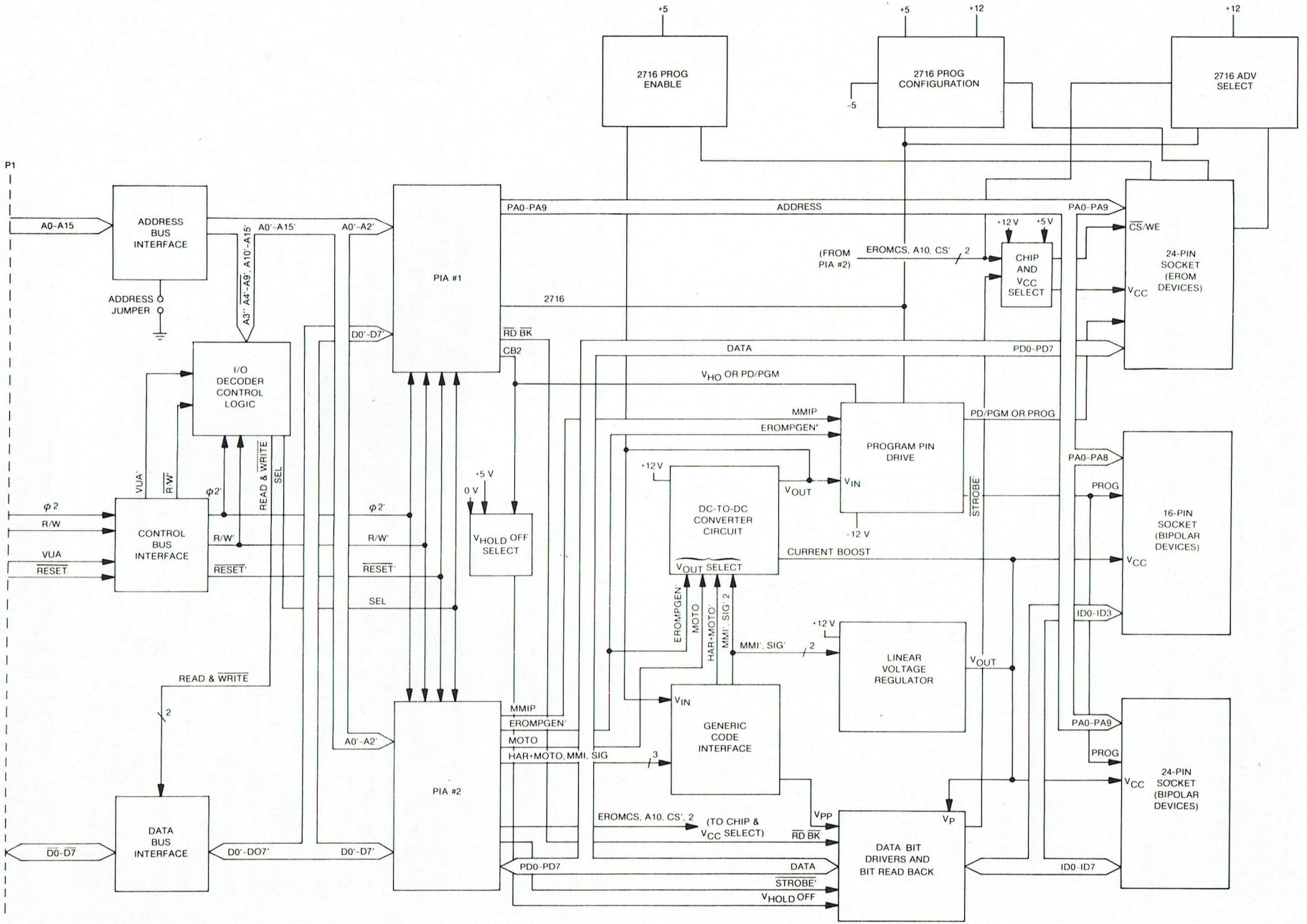


Figure 4-1. MEX68PP3 PROM Programmer III Module Block Diagram

REVISIONS						
REV	ECO NO.	ZONE	CHANGE	DATE	BY	ENGR
K	5174	–	RELEASED FOR PRODUCTION	1/4/77		

SYMBOL	QTY	PART NO.
U1,8	2	8T98
U2,3	2	MC6821
U4,5	2	8T26
U6,7,21,34	4	8T97
U9	1	74S86
U10,11	2	74S30
U12	1	74S02
U13	1	7410
U14,37,38	3	7404
U15,17,35,41	5	7406
U16,22,23	3	7402
U19	1	TL497
U20	1	1723
U24,25,31	3	7407
U27,28	2	MPQ2222
U29,33,48,50	4	MPQ3467
U30	1	Res 750
U32,33	2	7400
U36	1	Res 2.2K
U40,42	2	Res 2.2K
U43,44	2	7401
U45,49,26	3	Res 10K
U46,47	2	MC1108
U51	1	24 PIN EROM
U52	1	16 PIN BIPOLAR
U53	1	24PIN BIPOLAR

LAST REF NO. USED
U53
C28
CR19
R40
Q4

NOTES:
1. UNLESS OTHERWISE SPECIFIED:
   RESISTANCE VALUES ARE IN OHMS. CAPACITANCE VALUES ARE IN MICROFARADS. RESISTORS ARE 1/4 WATT.
2. MOUNT ALL COMPONENTS ON BOARD NO — 84DWG299X01
3. FOR PARTS LOCATION SEE ASSEMBLY DWG — 01DW1299X

ANY SPECIFICATIONS, DRAWINGS OR REPRINTS OR DATA FURNISHED TO BUYER OR SELLER SHALL REMAIN MOTOROLA'S PROPERTY, SHALL BE KEPT CONFIDENTIAL, SHALL BE USED FOR THE PURPOSE OF COMPLYING WITH MOTOROLA'S REQUESTS FOR QUOTATION OR WITH MOTOROLA PURCHASE ORDERS AND SHALL BE RETURNED AT MOTOROLA'S REQUEST. PATENT RIGHTS EMBODIED IN DESIGNS, TOOLS, PATTERNS, DRAWINGS, DEVICES, INFORMATION AND EQUIPMENT SUPPLIED BY MOTOROLA PURSUANT TO THIS REQUEST FOR QUOTATION OR PURCHASE ORDER AND EXCLUSIVE RIGHTS FOR THE USE IN REPRODUCTION THEREOF ARE RESERVED BY MOTOROLA.

MOTOROLA INC.
Discrete Semiconductor Division

TITLE: SCHEMATIC, PROM PROGRAMMER III

DRAWN BY	DATE
E/M DESIGN	10-11-77
CHECKED BY	DATE
ENGR APPROVAL	DATE

NEXT ASSEMBLY USED ON APPLICATION

SIZE E | CODE IDENT NO 04713 | DRAWING NO 68EW1299X | SCALE NONE | WEIGHT | SHEET 1 OF 2

FIGURE 4-2 "A" – MEX68PP3 EROM/PROM PROGRAMMER SCHEMATIC DIAGRAM

REVISIONS					
REV	ECO NO.	ZONE	CHANGE	DATE	BY	ENGR
K	0174	—	RELEASED FOR PRODUCTION	4-7-78	[illegible]

NOTES:
1. UNLESS OTHERWISE SPECIFIED:
   RESISTANCE VALUES ARE IN OHMS. CAPACITANCE VALUES ARE IN MICROFARADS. RESISTORS ARE ¼ WATT.
2. MOUNT ALL COMPONENTS ON BOARD NO. — 84DW6299X01
3. FOR PARTS LOCATION SEE ASSEMBLY DWG. — 01DW1299X

"ANY SPECIFICATIONS, DRAWINGS OR REPORTS, OR DATA FURNISHED TO BUYER OR SELLER SHALL REMAIN MOTOROLA'S PROPERTY, SHALL BE KEPT CONFIDENTIAL, SHALL BE USED FOR THE PURPOSE OF COMPLYING WITH MOTOROLA'S REQUESTS FOR QUOTATION OR WITH MOTOROLA PURCHASE ORDERS AND SHALL BE RETURNED AT BOTH BOLL'S REQUEST. PATENT RIGHTS EMBODIED IN DESIGNS, TOOLS, PATTERNS, DRAWINGS, DEVICES, INFORMATION AND EQUIPMENT SUPPLIED BY MOTOROLA PURSUANT TO THIS REQUEST FOR QUOTATION OR PURCHASE ORDER ARE EXCLUSIVE RIGHTS FOR THE USE OR REPRODUCTION THEREOF ARE RESERVED BY MOTOROLA."

MOTOROLA INC.
Discrete Semiconductor Division

TITLE: SCHEMATIC, PROM PROGRAMMER III

DRAWN BY E/M DESIGN	DATE 10-13-77	SIZE E	CODE IDENT NO. 04713	DRAWING NO. 63EW1299X
CHECKED BY	DATE	SCALE NONE	WEIGHT	SHEET 2 OF 2
ENGR APPROVAL	DATE			
NEXT ASSEMBLY	USED ON			
APPLICATION				

FIGURE 4-2 "B" — MEX68PP3 EROM/PROM PROGRAMMER SCHEMATIC DIAGRAM

CHAPTER 5

PROGRAMMER MODULE COMPONENTS

This chapter provides a list of components for the MEX68PP3 PROM Programmer III Module. Effectivities shown reflect the revision level of the Programmer bill of material at date of printing.

TABLE 5-1. PROM Programmer III Module Components

Reference Designation	Motorola Part Number	Description	Effectivity
	01DW1299X01	PROM Programmer III	
	84DW6299X01	Printed Wiring Board	A
R1,R2,R26, R34,R38	06SW-124A57	Resistor, fixed carbon, 2.2KΩ, 5%, 1/4W	C
R3	06SW-125D70	Resistor, fixed carbon, 1.0Ω, 10%, 1/2W	A
R4	06NW9602A43	Resistor, fixed film, 10.2KΩ, 1%, 1/8W	A
R5	18NW9603A25	Resistor, variable, 2KΩ, 10%, 1/2W	A
R6	06NW9602A25	Resistor, fixed film, 412Ω, 0.5%, 1/8W	B
R7	06NW9602A49	Resistor, fixed film, 750Ω, 1%, 1/8W	B
R8	06NW9602A46	Resistor, fixed film, 475Ω, 1%, 1/8W	A
R9	06NW9602A52	Resistor, fixed film, 1.07KΩ, 1%, 1/8W	A
R10	06NW9602A50	Resistor, fixed film, 845Ω, 1%, 1/8W	A
R11	06NW9602A48	Resistor, fixed film, 6.98KΩ, 1%, 1/8W	A
R12	06NW9602A51	Resistor, fixed film, 931Ω, 1%, 1/8W	A
R13,R35	06SW-124A25	Resistor, fixed carbon, 100Ω, 5%, 1/4W	C
R14	18NW9603A20	Resistor, variable, 100Ω, 10%, 1/2W	A
R15	06SW-124A43	Resistor, fixed carbon, 560Ω, 5%, 1/4W	A
R16	06NW9602A42	Resistor, fixed wirewound, 0.33Ω, 1%, 1W	A
R17	06NW9602A44	Resistor, fixed film, 1.5KΩ, 1%, 1/8W	A
R18	06NW9602A47	Resistor, fixed film, 5.9KΩ, 1%, 1/8W	A
R19,R20,R29.R36	06SW-124A49	Resistor, fixed carbon, 1KΩ, 5%, 1/4W	A
R21	06SW-124A73	Resistor, fixed carbon, 10KΩ, 5%, 1/4W	A
R22,R25,R40	06SW-124A41	Resistor, fixed carbon, 470Ω, 5%, 1/4W	B
R23	06SW-126A21	Resistor, fixed carbon, 68Ω, 5%, 1W	A
R24	06SW-124A35	Resistor, fixed carbon, 270Ω, 5%, 1/4W	A
R27,R30	06SW-124A45	Resistor, fixed carbon, 680Ω, 5%, 1/4W	A
R28	06SW-124A17	Resistor, fixed carbon, 47Ω, 5%, 1/4W	A
R31	06SW-124A57	Resistor, fixed carbon, 2.2KΩ, 5%, 1/2W	C

TABLE 5-1. PROM Programmer III Module Components (cont'd)

Reference Designation	Motorola Part Number	Description	Effectivity
R32	06SW-124A37	Resistor, fixed carbon, 330Ω, 5%, 1/4W	A
R33	06SW-124A56	Resistor, fixed carbon, 2.0KΩ, 5%, 1/4W	A
R39	06SW-124A01	Resistor, fixed carbon, 10Ω, 5%, 1/4W	C
R37	06SW-126A27	Resistor, fixed carbon, 120Ω, 5%, 1W	A
C1,C16,C23	23NW9618A24	Capacitor, electrolytic, 10 mfd., 25 Vdc, -10%, +75%	A
C2 through C15, C18,C19,C21	21NW9702A09	Capacitor, fixed ceramic, 0.1 mfd., 50 Vdc, 20%	A
C17	21NW9604A36	Capacitor,fixed ceramic,1.0 mfd.,200 Vdc, 20%	A
C20	23NW9618A28	Capacitor,electrolytic,500 mfd., 50Vdc, -10%, +150%	A
C22,C24	21NW9604A27	Capacitor,fixed ceramic, .820 mfd., 200 Vdc, 10%	A
C25	21NW9604A30	Capacitor,fixed ceramic, .039 mfd., 50 Vdc, 10%	A
C26,C27	21NW9604A29	Capacitor,fixed ceramic, .033 mfd., 50 Vdc, 10%	A
C28	215W994B031	Capacitor, mica, .150 mfd., 500 Vdc, 5%	A
L1	24AW1372X01	Inductor, Motorola Microsystems	A
Q1	48NW9611A17	Transistor, PNP, Power, MJE2955	A
Q1(Alt.)	48NW9611A11	Transistor, PNP, Power, MJE307K	A
Q2,Q3	48NW9611A01	Transistor, PNP, 2N4403, Case 29-1	A
Q4	48NW9610A01	Transistor, NPN, 2N4401	A
U1,U8	51NW9615C36	I.C., 8T98, Signetics	A
U2,U3	51NW9615D85	I.C., MC68B21P, Motorola	A
--	28NW9802B09	Socket (U2,U3), 40-pin, D1L, Burndy	A
U4,U5	51NW9615B35	I.C., 8T26, Signetics	A
U6,U7,U21,U34	51NW9615B71	I.C., 8T97, Signetics	A
U9	51NW9615C58	I.C., SN74586N, TI	A
U10,U11	51NW9615D93	I.C., SN7453ON, TI	A
U10,U11 (Alt.)	51NW9615C41	I.C., MC3016, Motorola	C
U12	51NW9615D32	I.C., SN7450 2N, TI	A
U13	51NW9615A38	I.C., MC7410P, Motorola	A
U14,U37,U38	51NW9615A35	I.C., MC7404P, Motorola	A
U15,U17,U18, U35,U41	51NW9615A36	I.C., MC7406P, Motorola	A
U16,U22,U23	51NW9615A34	I.C., MC7402P, Motorola	A
U19	51NW9615D17	I.C., TL497IN, TI	A

TABLE 5-1. PROM Programmer III Module Components (cont'd)

Reference Designation	Motorola Part Number	Description	Effectivity
U20	51NW9615C15	I.C., MC1723P, Motorola	A
U24,U25,U31	51NW9615B59	I.C., MC7407P, Motorola	A
U26,U45,U49	51NW9615A14	Resistor Network, 10KΩ,316B103,Allen Bradley	A
U27,U28	48NW9610A27	Transistor, NPN, MPQ2222, Motorola	A
U29,U39,U48,U50	48NW9611A24	Transistor, PNP, MPQ3467, Motorola	C
U29,U39,U48,U50 (Alt.)	48NW9611A16	Transistor, PNP, MHQ3467, Motorola	C
U30	51NW9626A26	Resistor Network, 750Ω, RA08-750N, National	A
U32,U33	51NW9615A32	I.C., MC7400P, Motorola	A
U36,U40,U42	51NW9626A25	Resistor Network, 2.2KΩ, 316B2B2, Allen Bradley	A
U43,U44	51NW9615A33	I.C., MC7401P, Motorola	A
U46,U47	51NW9615D18	I.C., MC1108P, Motorola	A
U51,U53	28NW9802B52	Socket, TS-61024, Robinson Nugent	A
U52	28NW9802B51	Socket, TS-31016, Robinson Nugent	A
CR1,CR2,CR3,CR9, CR10,CR12,CR13	48NW9816A03	Diode, Silicon, Signal, 1N914	A
CR4,CR5,CR6, CR15,CR18	48NW9607A01	Rectifier, 1N4001	A
CR7	48NW9608A09	Diode, Zener, Regulator	A
CR8,CR11,CR14, CR16,CR17	48NW9816A05	Diode, Silicon, Signal, MSD7000, Motorola	A
CR19	48NW9608A20	Diode, Zener, 1N5234	A
--	29NW9805A42	Terminal, Stud Turret, 29-15102A18, GED	A
--	29NW9805A46	Term, Wirewrap, 155-0019-01-03, Cambion	A
--	55NW9403A05	Ejector, Card, S-203, Scanbe	A

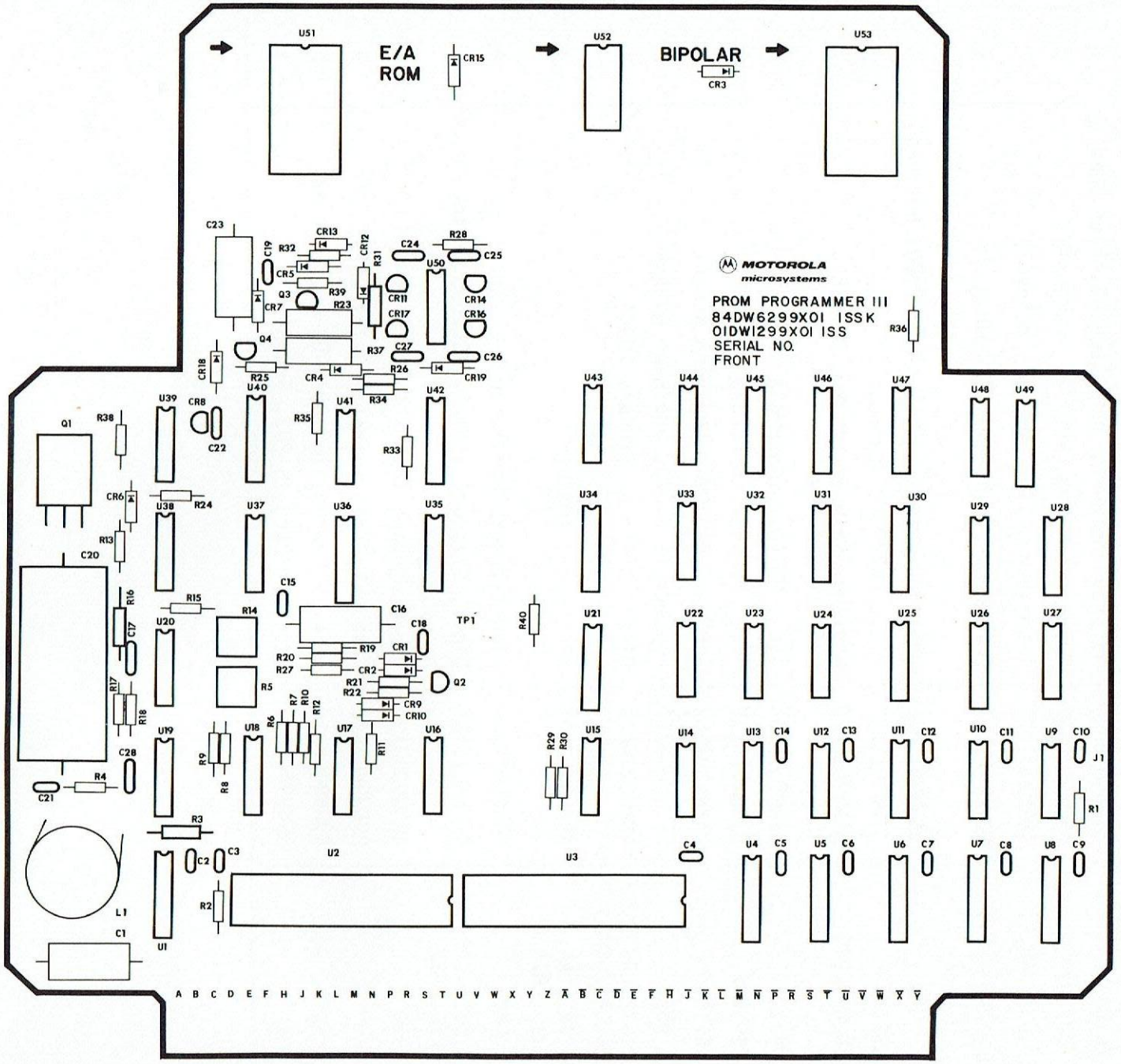


FIGURE 5-1. UNIVERSAL EROM/PROM PROGRAMMER MODULE COMPONENT LOCATION

APPENDIX A

PROGRAMMING EXAMPLES

The following examples illustrate typical applications of the PROM Programmer III Module. It is assumed in all examples that the Programmer software has been loaded into memory and started. These examples are based on the tape/EDOS version, running on an EXORciser. Important differences between other versions will be noted. Each Programmer command is terminated with a Carriage Return. All load and data addresses, and offsets, are in hexadecimal notation. All user input is underlined.

APPENDIX A-1

Program EROM/PROM from Tape or Disk

In this example, it is desired to program an MCM68708 1Kx8 EROM from data on tape or disk. The program to be put in the EROM/PROM normally resides from FC00 through FFFF.

P MCM68708 — This specifies the EROM/PROM to be programmed. When the prompt is displayed after this command has been accepted, the part may be inserted in the appropriate socket on the Programmer Module

One of the following three commands is used to load the data into memory:

To load memory from tape (tape/EDOS version) -

P TAPE 1400 — The tape should be inserted in the reader and the reader should be ready before this command is entered. The Programmer will pick up the values 1400 for nnnn, FC00 for xxxx, and FFFF for yyyy. The data will be loaded into memory from 1000 through 13FF.

To load memory from EDOS disk file (tape/EDOS version) -

P FILE TEST,1400 — This assumes the data to be loaded is in the file named TEST on drive 0. The Programmer will pick up the values 1400 for nnnn, FC00 for xxxx, and FFFF for yyyy. The program will be loaded into memory from 1000 through 13FF.

To load memory from MDOS disk file (MDOS version) -

P FILE TEST,3400 — This assumes that the data to be loaded is in the file named TEST.LO on drive 0. The Programmer will pick up the values 3400 for nnnn, FC00 for xxxx, and FFFF for yyyy. The program will be loaded into memory from 3000 through 33FF.

P PROG ,,0 — This programs the EROM/PROM from memory addresses xxxx+nnnn through yyyy+nnnn, beginning at EROM/PROM address 0. These are the same memory addresses the program was loaded into. If the EROM/PROM offset parameter zzz was already 0, then just PROG could have been entered. The device may be removed from the socket when the prompt is displayed again.

APPENDIX A-2

Duplicating a PROM

This example illustrates how to duplicate a PROM. The PROM used in this example is the MCM7640, a 512x8 bipolar PROM. The memory offset parameter nnnn is assumed to be 0. If it is not, the OFST command should be used to set it to 0.

P MCM7640	This specifies the PROM to be read (and programmed, since a different PROM is not specified before the PROG command is entered). When the prompt is displayed after this command has been accepted, the part may be inserted in the appropriate socket on the Programmer Module.
P READ 1000,11FF,0	This reads the PROM to memory addresses 1000 through 11FF, starting at PROM offset 0. If the PROM offset parameter zzz is already 0, it does not have to be reentered here. For the MDOS version, these addresses would have to be changed to 3000 and 31FF, respectively, in order to be above the Programmer in memory. When the prompt is displayed after this command has been successfully executed, the programmed part may be removed from the socket and an unprogrammed part inserted.
P PROG	This programs the PROM from the addresses entered during the READ command. The PROM may be removed from the socket when the prompt is displayed again.

APPENDIX A-3

Changing the Unprogrammed Bits of a PROM

This example demonstrates how to change a bipolar PROM that has already been programmed. The bit, or bits, to be changed must not have been programmed, since a bit can change state only once in a bipolar PROM. Some EROM manufacturers do not recommend programming only a portion of an EROM. Instead, they recommend that the EROM be erased and complete reprogrammed. The PROM used in this example is an MMI6300, a 256x4 bipolar PROM. It is desired to change the contents of location A3 of the PROM from 5 to 4 (MMI PROMs program from a 1 to a 0). The memory offset parameter nnnn is assumed to be 0. The OFST command should be used to set it to 0 if it is not.

P MMI6300

This specifies the PROM to be read and programmed. When the prompt is displayed after this command has been accepted, the part may be inserted in the appropriate socket on the Programmer Module.

>P READ 1000,1000,A3

This reads the PROM location to be changed (A3) to memory address 1000. For the MDOS version, the memory address used should be 3000 instead of 1000, in order to be above the Programmer. The symbol > appearing before the prompt indicates that the PROM contents will be read into the four least significant bits of memory.

>P EXIT

This returns control to the system monitor so that the data can be changed in memory.

```
BKPT ERROR
P-cccc X-dddd A-ee B-ff C-gg S-hhhh
*1000/E5 4
```

The system monitor is used to change the data. In the MDOS version, location 3000 would be changed. The four most significant bits of memory contain what they did before the READ command, which is not important in this example.

*407;G

This restarts the Programmer at the warm start entry point, thus retaining the Programmer parameters. For the MDOS version, ;P should be entered, instead of 407;G.

```
>P PROG
CHNG? Y
```

Location A3 has already been programmed once, so the Programmer prompts with the question CHNG?.
The Y response causes the Programmer to continue with the PROG command and program location A3. The PROM may be removed from the socket when the prompt is displayed again.

APPENDIX A-4

Possible Error Modes

This example describes some of the possible problems that a user may encounter and how to remedy the problems.

Assume that a user file called BURN exists in memory. Also assume that the present user is not aware that BURN can overwrite the Programmer program. If the user enters:

```
P FILE BURN
ADDR?          (Error message; see Appendix B)
```

After being informed by the error message ADDR?, the user decides to read the file "BURN" with a 3000 offset and enters:

```
P FILE BURN 3000
SYNTAX?        (Error message; see Appendix B)
```

The required comma, between BURN and 3000, was missing. The user then tries:

```
P FILE BURN,3000
P              (OK, so far)
```

The user desires to see the contents of memory, from the beginning of the file BURN, for 20 locations. The following entry is made:

```
P SHWM,3120
```

The response:

```
3100  xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
3110  xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
 ↓      ↓ ↓          (Memory Contents)              ↓
 ↓      ↓ ↓                                         ↓
6120  xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
P
```

The user was not aware that the listing address should be used. The correct entry is:

```
P SHWM,110
3100  xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
3110  xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
P
```

The user desires to program 256 (100 hex) locations of a Signetics 82S2708 1Kx8 bipolar PROM. PROM locations 100 through 1FF are to be programmed with the contents of memory locations 200 through 2FF. The following entries are made:

```
P 82S2708  (The part is a Signetics 82S2708)
P STAT     (The user wants the Programmer status)
```

The response is:

```
82S2708   1024x8    BIPOLAR
XXXX    YYYY    NNNN    AAA    ZZZ
0100    0110    3000   0400   0000
P
```

where:

XXXX=0100 (the starting address)
YYYY=0110 (the ending address)
NNNN=3000 (the memory offset)
AAA=0400 (the number of device locations)
ZZZ=0000 (the device offset)

The user now places the device in the BIPOLAR socket and enters:

```
P PROG 200,2FF,100
```

P The P returns, indicating that the device programmed successfully. The user now wishes to see the data in the device. The following entry is made:

```
P SHWP
```

The response is:

```
SIZE?
```

P The user examines the status by entering:

```
P STAT
```

The response is:

```
82S2708
XXXX    YYYY    NNNN    AAA    ZZZ
0200    02FF    3000   0400   0100
```

The problem is that the Programmer has been instructed to display 400 hex locations from a device address of 100 hex. Since this is 2FF locations instead of 400, the SIZE? printout is displayed. The user then enters:

```
P SHWP 200,
0100  xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
0110  xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
 ↓      ↓  ↓                                      ↓
 ↓      ↓  ↓                                      ↓
02F0  xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
P
```

The device is now unprogrammed from 000 to 00FF. The contents of the user file BURN, from 200 to 2FF, is in the device from 100 to 1FF. Locations 200 to 3FF, in the device, are unprogrammed.

APPENDIX B

Error Messages

The following messages can occur in all versions of the PROM Programmer III Module.

WHAT?	This indicates that an invalid command or unknown command or PROM specification was entered.
SYNTAX?	This indicates that an invalid character occurred in the parameters of the last command, or that an invalid drive number was entered in the EDOS FILE command. The command was not executed.
ADDR?	This occurs when the memory address to be stored by the TAPE, FILE, or READ command is between address 0000 and the last address of the Programmer. The command is then aborted. Refer to the memory map in Appendix E to determine the last address of the Programmer.
CKSM?	This indicates that a checksum error occurred during the TAPE or EDOS FILE command. The command is aborted when the checksum error occurs.
?	This indicates that a non-valid character was read, or that an end of tape or end of file was detected during a record, during the TAPE, or EDOS FILE commands, respectively. The command is aborted when this error occurs.
NO CHNG	This occurs during the TAPE or EDOS FILE commands when the location to be loaded does not change. The command is then aborted. This occurs when trying to load into ROM or non-existent memory.
NO CHNG mmmm pp	When a bit in a bipolar PROM cannot be programmed, this message is printed and the PROG command is aborted; mmmm is the PROM address of the word that will not program and pp is the contents of that location.
PWR?	This indicates that the +5 volt supply for bipolar PROMs would not come up to a logic 1 level. This condition can be detected by any command that reads or programs the PROM: CHCK, PROG, READ, SHWP, and VERF. The command is aborted when this condition occurs. Typically, this error is caused by improper insertion of the PROM in the socket, or by a defective PROM. This error message can also result when the Programmer Module address has been changed, but the program has not been modified to reflect the changed address.

CHNG? This occurs when the portion of the EROM/PROM to be programmed reads as if it were already programmed. Typing Y will cause the PROT command to continue and attempt to program the EROM/PROM. Typing any other character will cause the PROG command to abort. If the EROM/PROM had not been programmed previously, this message could result from improperly inserting the EROM/PROM in the socket.

SIZE? This occurs during the PROG, READ, and VERF commands when the difference between the ending address yyyy and the starting address xxxx is greater than the difference between the PROM size and the PROM offset zzz. Also, this message occurs during the CHCK and SHWP commands when the number of PROM locations to check or show, aaa, plus the EROM/PROM offset, zzz, is greater than the EROM/PROM size. The command is aborted.

OFST PR
mmmm pp This is printed during the CHCK command when a PROM location has been programmed; mmmm is the PROM address of the programmed location and pp is the contents of that location. The heading "OFST PR" is printed only once during the command. The EROM/PROM information is printed up to eight times per line.

OFST PR MM
mmmm pp qq This is printed during the PROG command, while verifying, and during the VERF command, when the EROM/PROM and memory do not compare; mmmm is the EROM/PROM address of the miscomparison, pp is the PROM contents, and qq is the memory contents. The heading "OFST PR MM" is printed only once during the command. The EROM/PROM and memory information is printed up to six times per line.

The EDOS version of the PROM Programmer III Module may also give the following error messages when the FILE command is executed:

NO SUCH FILE This results when the file name in the FILE command is not found on the disk drive specified.

DISK NOT READY When the disk drive specified in the FILE command is not ready, this message is printed. The drive may be not ready because a disk is not inserted in the drive, or the drive door is open.

MEDIA ERROR This is printed when the disk specified in the FILE command cannot be read or a CRC error is detected.

The MDOS version FILE command may also print MDOS Unified I/O and MDOS PROM I/O errors. Refer to the MDOS manual for the cause of these error messages.

ADDENDUM
TO THE
MEX68PP3
PROM PROGRAMMER III MODULE USER'S GUIDE

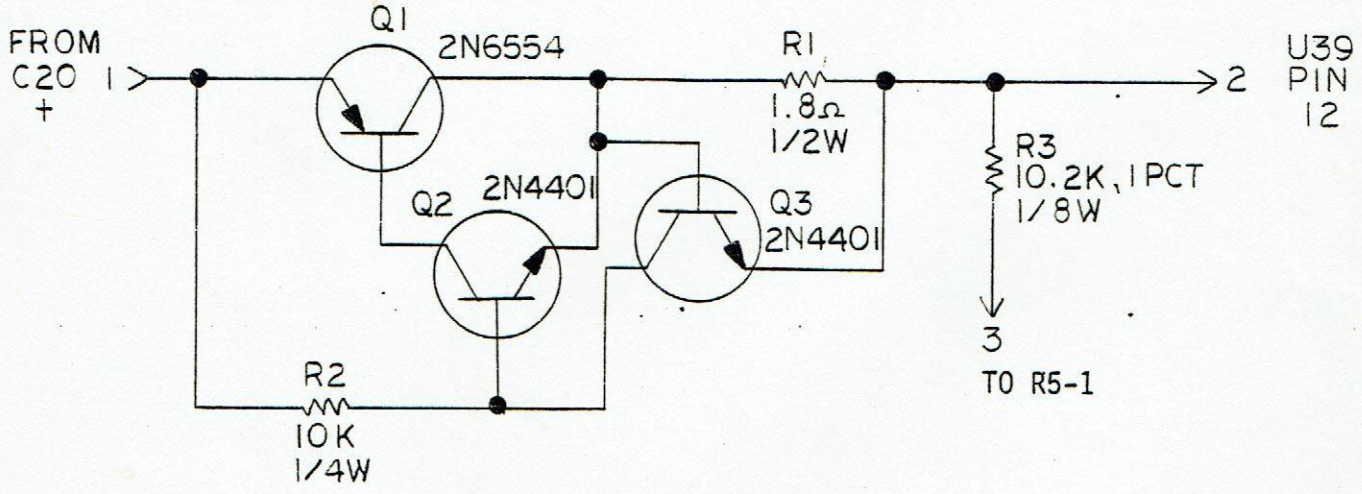
Due to the possibility of a catastrophic failure when a defective PROM is used in PP3, a modification has been incorporated. This addition consists of a current limiter circuit employing 3 transistors, and is shown in figure below.

This modification does not change the operation of the programmer, nor make a noticeable difference in its performance.

Typically the yield for programming a batch of PROMS will be about 95%. The percentage varies from manufacturer to manufacturer, and from batch to batch, however. It is sometimes possible to correct a failure to program by repeating the process again or even a third time. If repeated tries do not result in an error free verification, it is an indication that the part is unprogrammable on the PP3, and another part should be used.

If a higher than normal percentage of failures occur with a batch of PROMS, it is adviseable to try another batch before suspecting a defective PP3.

Pin 5 of U39 and pin 11 of U40 have been moved from the output of the programmable voltage regulator (U19) directly to +12v. This permits programming of Monolithic memories (MMI) PROMs with normal yields.



PPIII/CURRENT REG. BOARD SCHEMATIC

Resistor R4 on PPIII has been relocated to the current regulator board and is marked R3. The location R4 is now occupied by a 35v Zener IN4752, which is connected between R5-1 and C20+.

ADDENDUM
TO THE
MEX68PP3
PROM PROGRAMMER III MODULE SUPPLEMENT

In Chapter 3, Page 3-2, change the following paragraph as shown below:

FROM

EROM/PROMs may be inserted or removed any time the Programmer is displaying its prompt. EROM/PROMs should not be removed or inserted while a command is being executed. IN ALL CASES THE EROM/PROM MUST BE INSERTED IN THE PROPER SOCKET AS MARKED ON THE BOARD.

TO

CAUTION

EROM/PROMs must not be inserted before power is applied to the programmer or before the load and startup procedure is performed. THE DEVICE MAY BE INSERTE ONLY AFTER ENTERING THE DEVICE TYPE CODE. Thereafter, EROM/PROMs of the same device type may be inserted or removed any time the Programmer is displaying i prompt. When changing from one device type to another, the new device must no be inserted without first calling the new device-type code. EROM/PROMs should not be removed or inserted while a command is being executed. IN ALL CASES, TH EROM/PROM MUST BE INSERTED IN THE PROPER SOCKET, AS MARKED ON THE BOARD.

ADDENDUM

TO

MEX68PP3 PROM PROGRAMMER III MODULE SUPPLEMENT

Due to a problem in the executive program supplied with the MEX68PP3 PROM Programmer III Modules, attempts to program the Motorola bipolar PROM's listed by entering the identifying part numbers provided in Table 1-2 of the above supplement will result in destruction of the devices. Until revised software is available, the Motorola devices can be programmed by identifying them by their corresponding Harris part numbers, as shown in the following table.

To Program Motorola Part Number	Enter Harris Part Number	Description
MCM7640	HM-7640	512 x 8 Bipolar
MCM7641	HM-7641	512 x 8 Bipolar
MCM7680	HM-7680	1K x 8 Bipolar
MCM7681	HM-7681	1K x 8 Bipolar

APPENDIX C

Adapting Programmer Software to Your System

The following is a description of the modifications to the Programmer software required by various systems. Normally, these modifications are made once, and the modified version would be used.

The tape/EDOS version of the PROM Programmer may also be put into EROM/PROM, and run from EROM/PROM, provided the following conditions are met:

1. The 3K block from hex 400 through hex FFF is put in EROM/PROM.
2. All required modifications have been made to the Programmer.
3. The Programmer is used to program itself into EROM/PROM on the system in which the EROM/PROMs will be used.
4. The EROM/PROM version will be run only in the system in which it was programmed or in an identical system (same monitor and no clock speed or Programmer Module address changes).
5. The EDOS FILE command will not be used.

WHEN MODIFICATION IS NECESSARY

The Programmer must be modified if any of the following statements apply:

1. The monitor used with the Programmer is not EXbug I.
2. The system clock frequency is greater than 1.0 MHz or less than or equal to 0.7 MHz.
3. The Programmer Module is addressed at hex BC08 instead of EC08.

INCORPORATING CHANGES

The following procedures are recommended for incorporation in the various versions of the PROM Programmer.

Tape version

To modify the tape version, the tape originally received should be loaded into memory and the modifications should be made in memory; a new tape should be punched from hex location 400 to the address contained in HIAD and HIAD+1. The address contained in these two locations is the address of the first available byte past the end of the Programmer. The relative address of HIAD is given in another section. The new tape is loaded in place of the original tape when the PROM Programmer is used. This procedure leaves the EDOS version FILE command, which resides below hex 400, off the tape.

EDOS version

The EDOS version is most easily modified by first creating an assembly language source file of FCBs containing the required modifications ORGed at the appropriate addresses. The source file is then assembled to create a corresponding object file. Finally, this object file is merged to the end of the original PROMP file. Refer to the EDOS manual for information about EDOS commands. The file resulting from the merging of PROMP and the modification file is loaded in place of PROMP when the Programmer is used.

MDOS version

Modification of the MDOS version can be accomplished by using the MDOS PATCH command to change the necessary locations in the file PROMPROG.CM. Since the PATCH command automatically uses the lowest address of the patched file as its offset, the relative addresses of the modified locations given in the next section can be used directly in the PATCH command. Refer to the MDOS manual for further information about the MDOS PATCH command. The patched PROMPROG.CM file is loaded when the PROM Programmer is used.

WHAT TO CHANGE

The following information describes the PROM Programmer locations that must be changed to adapt the Programmer to various systems. The relative addresses given in this section must be added to a base address in order to determine the actual memory address to be changed. For the tape/EDOS version, this base address is hex 400. The base address of the MDOS version can be determined by using the MDOS PATCH command to display the lowest address of the Programmer, which is the MDOS version base address. The contents given for the following locations are in hex.

If the Programmer is used with a system monitor other than EXbug I, location MONITR listed below will have to be changed. Additionally, if the monitor is an unsupported monitor (one not listed in the MONITR description), then the locations INPUT, OUTPUT, EXIT, ACIA, and ACIACD described below should be changed and a 0 put in location MONITR. Also, the CKBRK jump described below may require changing. If the unsupported monitor does not use an ACIA for console communications, the user will have to supply his own routine to check for the control-W character and a break condition, or replace the jump instruction with a CLC and an RTS instruction. The RDRIN, RDROFF, and RDRON jumps may also need changing in this case since these routines expect an ACIA to be used for the terminal interface.

If the system clock frequency is less than or equal to 0.7 MHz or greater than 1.0 MHz, location CLCK must be changed.

If the Programmer Module address is not hex EC08, then the location PIA must be changed.

In order to read S record object from a device other than the system console reader using the TAPE command, the user can change the jump instructions at RDRIN, RDROFF, and RDRON to jump to corresponding routines that he supplies to interface to his device.

RED ADDR	LABEL	DESCRIPTION
0000-0003		These four locations contain version and revision number information.
0004-0006		These three locations contain a jump instruction to the PROM Programmer's cold start point.
0007-0009		These three locations contain a jump instruction to the PROM Programmer's warm start point.

000A-000B HIAD These two locations contain the address of the first available memory byte past the end of the Programmer. This address is used by the Programmer during commands which load memory. Memory address less than the address in these locations will not be loaded.

000C BTRY This location contains the primary bipolar program try count. The default value is 6.

000D BRETRY This location contains the secondary bipolar program try count. The default value is 2.

000E ETRY This location contains the EROM program repetition count. The default value is hex C8 (decimal 200).

000F CLCK This location contains a value indicating the system clock frequency. The value must be one of the values listed in the following table.

CLCK	CLOCK FREQUENCY=F				
FF	0.5 MHz	=	F	=	0.7 MHz
00	0.7 MHz		F	=	1.0 MHz - DEFAULT
01	1.0 MHz		F	=	1.5 MHz
02	1.5 MHz		F	=	2.0 MHz

Running the system at a clock frequency at the lower end of a range will result in longer programming times, but the PROM Programmer's specifications will not be violated.

0010-0011 PIA These two locations contain the lowest address of the PIAs on the PROM Programmer Module. The default address is hex EC08. The address in these locations must be changed to reflect any change in the PIA's addresses. If the jumper is installed on the Programmer to change the base PIA address to hex BC08, the address in these locations must be changed to BC08, also.

0012 MONITR This location contains a value indicating the system monitor in use and whether the disk commands are valid. If the most significant bit of this location is set to 1, then the DISK and FILE commands are valid. If the most significant bit is reset to 0, then the DISK and FILE commands are invalid. The other seven bits of this location are used as a binary number to indicate the sytem monitor as listed in the following table.

MONITR	SYSTEM MONITOR
00	EXbug I (or user) - DEFAULT
01	Reserved for future use
02	MICRObug
03	MINIbug III
04	MINIbug II

If an incorrect or invalid value is used for MONITR, the PROM Programmer might not respond or might respond incorrectly when started. When the PROM Programmer is used with

MINIbug II or III, the Programmer Module address will probably have to be changed to hex BC08 due to the system memory map. Refer to the hardware section of this manual for instructions on changing the board address. Do not forget to change location PIA listed above when the board address is changed. If an unsupported monitor is used with the Programmer, the MONITR location should be set to 0 and the following nine entries changed as required.

0013-0015 RDRON This is a jump instruction, executed by the TAPE command, that goes to a subroutine to turn on the console reader. This command uses the ACIA address contained in location ACIA described below.

0016-0018 RDROFF This is a jump instruction, executed by the TAPE command, that goes to a subroutine to turn off the console reader. The subroutine uses the ACIA address contained in location ACIA described below. After turning the reader off, this subroutine reads characters using the RDRIN routine until a timeout is detected.

0019-001B RDRIN This is a jump instruction, executed by the TAPE command, that goes to a subroutine that gets a character from the console reader in the A accumulator with the parity bit reset. Carry is returned set if an end of tape (timeout) is detected; otherwise, carry is returned cleared. The subroutine uses the ACIA address contained in location ACIA described below.

001C-001E CKBRK This is a jump instruction to a subroutine that checks for the control-W character and a break condition (framing error) from the terminal interface ACIA. If a control-W is entered, control is not returned from the subroutine until some other character is entered or a break is detected. If a break is detected, carry is returned set; otherwise, carry is returned cleared. The subroutine uses the ACIA address contained in location ACIA described below. This subroutine must preserve the X register and the B accumulator.

001F-0021 INPUT These locations contain a JSR instruction to a system console input character subroutine internal to the Programmer. This input subroutine uses the ACIA defined at location ACIA described below. After receiving a character, it is echoed to the console using the OUTPUT routine defined below due to the JSR instruction. For an unsupported monitor that does not use an ACIA, location MONITR should be 0 and the appropriate monitor address should be patched into this JSR instruction. The console input subroutine should not echo the input character back to the console. It should return the character in the A accumulator. If the character is printed as it is sent or the user's input subroutine does echo the character, the JSR instruction should be changed to a jump instruction so that after getting the character, control will not be passed to the next instruction listed below which is a jump to the system monitor's output subroutine.

0022-0025 OUTPUT These three locations contain a jump instruction to the system monitor's console output subroutine. Initially, the jump instruction contains the address for the EXbug I subroutine. However, if the MONITR number is non-zero, the address in the jump instruction will be changed for the designated monitor after the Programmer is started at its cold start point. For an unsupported monitor, location MONITR should be 0 and the appropriate address should be patched into this jump instruction. The console output subroutine should send the ASCII character in the A accumulator to the system console. This subroutine must preserve the X register and the A and B accumulators. This subroutine is also used by the CKBRK routine to send a null to the console when a break is detected.

0025-0027 EXIT These locations contain an instruction that will result in control being returned to the system monitor. Initially, these locations contain an SWI instruction, followed by two NOP instructions for EXbug I in the tape/EDOS version. With MDOS, a more complicated procedure must be followed to return properly to Exbug. Therefore, in the MDOS version these locations contain a jump instruction to a routine which returns to EXbug. If the MONITR number is non-zero, a jump instruction to the designated monitor's entry address will be placed in these locations once the Programmer is started at its cold start point. For an unsupported monitor, location MONITR should be 0; and if an SWI instruction will not return control to the monitor, a jump instruction containing the monitor reentry address should be patched into these locations.

0028-0029 ACIA These locations contain the address of the ACIA which interfaces to the system console. Initially, these locations contain the address of the EXbug I ACIA. If the MONITR number is non-zero, the appropriate ACIA address for the designated monitor will be put in these locations after the Programmer started at its cold start point. For an unsupported monitor, location MONITR should be 0 and the address of the system monitor's ACIA should be patched into these locations. The ACIA address is used by the INPUT, CKBRK, RDRON, RDROFF, and RDRIN routines.

002A-002B ACIACD These two locations contain the address of a location which contains the normal control word used for the ACIA. Initially, these locations contain the correct address for EXbug I. However, if the MONITR number is non-zero, the appropriate address for the designated monitor will be put in these locations after the Programmer is started at its cold start point. The ACIA control word is used to keep the correct word length and number of stop bits while changing the ACIA RTS line to control a relay modified TTY reader. This is done in the Programmer's RDRON and RDROFF subroutines. For an unsupported monitor, location MONITR should be 0 and these two locations should be patched to contain the address of a location which contains the ACIA control word. This cannot be the ACIA address since the ACIA control register is a write only register.

APPENDIX D

EROM/PROM Program Flow Description

This section describes the procedure followed by the Programmer in the actual programming of an EROM/PROM.

BIPOLAR PROMs

This section describes the portion of the Programmer which programs a bit in a bipolar PROM. (Bipolar PROMs are programmed a bit at a time). This section helps the user to determine which Programmer variables to change in order to improve programming success with difficult-to-program bipolar PROMs. This description begins at the point where a bit has been found which must be programmed. The labels used for maximum loop count parameters are described in Appendix C.

```
DO until bit programmed or a maximum of BRETRY times
     DO until bit programmed or a maximum of BTRY times
          attempt to program bit
     END DO
     IF bit did not program THEN
          turn power to part off
          wait 5 seconds
     END IF
END DO
IF bit did not program THEN
      print error message
      abort PROG command
END IF
```

EROMs

This section describes the portion of the Programmer which programs EROMs. The duration of the program pulse applied to the EROM at each location is approximately 0.5 milliseconds when the clock frequency is at the high end of one of the ranges specified in the CLCK location description.

```
DO for ETRY times
     LOCATION = start of EROM area to be programmed
     DO until LOCATION = end of area to program + 1
          program location
          LOCATION = LOCATION + 1
     END DO
END DO
```

APPENDIX E

Memory Maps

The following memory maps describe the memory locations of the tape/EDOS and MDOS versions of the PROM Programmer software. All addresses are in hex.

Tape/EDOS version

```
0000
     reserved for disk parameters
0020
     Programmer parameters
     command input buffer
0010
     EDOS file command
     stack
0380
     reserved for MICRObug RAM
0400
     PROM Programmer
     patch area
address contained in HIAD (typically less than 1000)
HIAD is described in Appendix C.
```

MDOS version

```
0000
     reserved for disk parameters
0020
     Programmer parameters
     command input buffer
0100
     resident MDOS
BASE supplied by RLOAD (depends on the version of MDOS in use)

This address can be determined using the MDOS PATCH command on the
PROMPROG.CM file

     PROM Programmer
     patch area
     MDOS FILE command
     stack

Address contained in HIAD (typically less than 3000)
HIAD is described in Appendix C.
```

MOTOROLA INC.

Integrated Circuits Division

Microsystems • 3102 North 56th Street • Phoenix, Arizona 85018

11731-1 PRINTED IN USA (2/79) MPS